	Hits	Search Text	DBs	Time Stamp
1	6	("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.	USPAT; US-PGPUB	2003/07/10 17:59
2	1	(("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (savepoint or save?point or checkpoint or check?point or syncpoint or sync?point)	USPAT; US-PGPUB	2003/07/10 18:00
3	5	(("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (rollback or roll?back or (roll$3 adj back) or undo)	USPAT; US-PGPUB	2003/07/10 18:00
4	1	((("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (savepoint or save?point or checkpoint or check?point or syncpoint or sync?point)) and ((("6275832" "6334134" "6125370" "6185699" "6098075" "5897641").did.) and (rollback or roll?back or (roll$3 adj back) or undo))	USPAT; US-PGPUB	2003/07/10 18:08
5	15	(transaction near6 lock$3) and ((savepoint or save?point or checkpoint or check?point or syncpoint or sync?point) near6 lock$3) and (rollback or roll?back or (roll$3 adj back) or undo)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 18:15

	Hits	Search Text	DBs	Time Stamp
6	15	((transaction near6 lock$3) and ((savepoint or save?point or checkpoint or check?point or syncpoint or sync?point) near6 lock$3) and (rollback or roll?back or (roll$3 adj back) or undo)) and @ad<=20010628	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:05
7	327	(transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or ((subtransaction or sub?transaction or steps!) near2 (point or portion or snapshot or	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:10
8	356	(transaction with lock$3) and ((savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transaction or steps!) with	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:12
9	43	((transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or ((subtransaction or sub?transaction or steps!) near2 (point or portion or snapshot or snap?shot))))) and ((transaction with lock$3) and ((savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transaction or steps!) wi	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:12

	Hits	Search Text	DBs	Time Stamp
10	21	(((transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or ((subtransaction or sub?transaction or steps!) near2 (point or portion or snapshot or snap?shot))))) and ((transaction with lock$3) and ((savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transaction or steps!) with lock$3))) and ((rollback or roll?back or (roll$3 adj back) or undo or revers$3) near5 (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transactio	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:13

	Hits	Search Text	DBs	Time Stamp
11	19	((((transaction with (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or ((subtransaction or sub?transaction or steps!) near2 (point or portion or snapshot or snap?shot))))) and ((transaction with lock$3) and ((savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transaction or steps!) with lock$3))) and ((rollback or roll?back or (roll$3 adj back) or undo or revers$3) near5 (savepoint or save?point or "save point" or checkpoint or check?point or "check point" or subtransaction or sub?transaction or steps!))) and	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:50
12	4	((savepoint or save?point or "save point") near4 (rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out))) same (lock$3 or x?lock or u?lock or s?lock)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:56
13	24	transaction with (savepoint or save?point or "save point")	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:57

	Hits	Search Text	DBs	Time Stamp
14	28	(rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (savepoint or save?point or "save point")	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:58
15	9	((rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (savepoint or save?point or "save point")) and (releas$3 near2 (lock or x?lock or u?lock or s?lock or latch))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 19:59
16	9	(((rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (savepoint or save?point or "save point")) and (releas$3 near2 (lock or x?lock or u?lock or s?lock or latch))) and @ad<=20010628	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:04
17	2749	(707/200 707/201 707/202 707/8).ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:04
18	309	((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:05

	Hits	Search Text	DBs	Time Stamp
19	48	(((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))) and ((rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:07
20	21	((((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))) and ((rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))) and (releas$3 with (lock or x?lock or u?lock or s?lock or latch))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:07

	Hits	Search Text	DBs	Time Stamp
21	19	(((((707/200 707/201 707/202 707/8).ccls.) and (transaction with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))) and ((rollback or roll?back or (roll$3 adj back) or backout or back?out or (back$3 adj out)) with (steps! or savepoint or save?point or "save point" or checkpoint or check?point or "check point"))) and (releas$3 with (lock or x?lock or u?lock or s?lock or latch))) and @ad<=20010628	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/10 20:07
22	8	("5630124" "5452445" "5490271" "6105025" "6047285" "5280612" "5806075" "5737601").did.	USPAT; US-PGPUB	2003/07/11 11:14
23	1	((rollback or roll?back or (roll$3 adj back)) near4 (savepoint or save?point or "save point" or intermediate or partial)) same ((releas$3 near1 lock) or unlock$3)	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2003/07/11 11:16
24	264	(indrajit bertino jajodia mancini).in.	USPAT; US-PGPUB	2003/07/11 11:52
25	0	((indrajit bertino jajodia mancini).in.) and commit	USPAT; US-PGPUB	2003/07/11 11:52
26	7	((indrajit bertino jajodia mancini).in.) and transaction	USPAT; US-PGPUB	2003/07/11 11:53
27	28	(watts and julie).in.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 14:18

	Hits	Search Text	DBs	Time Stamp
28	12	((watts and julie).in.) and (savepoint or save?point or "save point")	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 14:19
29	2	(database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll$3 adj back)) and (lock$3 with (savepoint or save?point))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 16:12
30	46	(database or data?base) and transaction and (savepoint or save?point or "save point") and (rollback or roll?back or (roll$3 adj back))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 16:13
31	44	((database or data?base) and transaction and (savepoint or save?point or "save point") and (rollback or roll?back or (roll$3 adj back))) not ((database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll$3 adj back)) and (lock$3 with (savepoint or s	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 16:13

	Hits	Search Text	DBs	Time Stamp
32	11	(((database or data?base) and transaction and (savepoint or save?point or "save point") and (rollback or roll?back or (roll$3 adj back))) not ((database or data?base) and transaction and (savepoint or save?point) and (rollback or roll?back or (roll$3 adj back)) and (lock$3 with (savepoint or save?point)))) and (lock$3 with (transaction or read or shared) with (maintain$3 or commit or committing or committed))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/23 16:14
33	7	((savepoint or save?point or "save point" or (partial adj (rollback or roll?back))) same lock$3) and transaction and ((repeatable adj read) or ((read or maintain) with lock$3 with (commit or committed or committing)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/24 13:20
34	3142	(707/200 707/201 707/202 707/8).ccls.	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/24 13:21
35	57	((707/200 707/201 707/202 707/8).ccls.) and (savepoint or save?point or "save point" or (partial adj (rollback or roll?back)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/24 13:22

	Hits	Search Text	DBs	Time Stamp
36	15	(((707/200 707/201 707/202 707/8).ccls.) and (savepoint or save?point or "save point" or (partial adj (rollback or roll?back)))) and ((repeatable adj read$3) or (read$3 same lock$3 same (commit or committed or committing)))	USPAT; US-PGPUB; EPO; JPO; IBM_TDB	2004/02/24 13:23

Search Results

Search Results for: **[(((savepoint or save?point or "save point" or (partial <near/1> (rollback or roll?back))) <sentence> lock*) and (transaction <paragraph> lock*) and ((repeatable <near/1> read*) or (lock* <sentence> (read or maintain* or duration) <sentence> (commit or committing or committed))))<AND>(meta_published_date <= 06-01-2001)]**

Found **16** of **127,132 searched.**

Search within Results

> Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 16 of 16 short listing

1 An advanced commit protocol for MLS distributed database systems 100%
Indrajit Ray , Elisa Bertino , Sushil Jajodia , Luigi Mancini
Proceedings of the 3rd ACM conference on Computer and communications security January 1996

2 Analysis of locking behavior in three real database systems 100%
Vigyan Singhal , Alan Jay Smith
The VLDB Journal — The International Journal on Very Large Data Bases February 1997
Volume 6 Issue 1
Concurrency control is essential to the correct functioning of a database due to the need for correct, reproducible results. For this

reason, and because concurrency control is a well-formulated problem, there has developed an enormous body of literature studying the performance of concurrency control algorithms. Most of this literature uses either analytic modeling or random number-driven simulation, and explicitly or implicitly makes certain assumptions about the behavior of transactions and t ...

3 Ordered shared locks for real-time databases 100%

Divyakant Agrawal , Amr El Abbadi , Richard Jeffers , Lijing Lin

The VLDB Journal — The International Journal on Very Large Data Bases January 1995

Volume 4 Issue 1

We propose locking protocols for real-time databases. Our approach has two main motivations: First, locking protocols are widely accepted and used in most database systems. Second, in real-time databases it has been shown that the blocking behavior of transactions in locking protocols results in performance degradation. We use a new relationship between locks called ordered sharing to eliminate blocking that arises in the traditional locking protocols. Ordered sharing eliminates blocking of read ...

4 Efficient concurrency control in multidimensional access methods 100%

Kaushik Chakrabarti , Sharad Mehrotra

ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international conference on Management of data June 1999

Volume 28 Issue 2

The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a “commercial-strength” database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

5 Concurrency control: methods, performance, and analysis 100%
Alexander Thomasian
ACM Computing Surveys (CSUR) March 1998
Volume 30 Issue 1

6 Concurrency and recovery in generalized search trees 100%
Marcel Kornacker , C. Mohan , Joseph M. Hellerstein
ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data June 1997
Volume 26 Issue 2
This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

7 Recovery protocols for shared memory database systems 100%
Lory D. Molesky , Krithi Ramamritham
ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data May 1995
Volume 24 Issue 2
Significant performance advantages can be gained by implementing a database system on a cache-coherent shared memory multiprocessor. However, problems arise when failures occur. A single node (where a *node* refers to a processor/memory pair) crash may require a reboot of the entire shared memory system. Fortunately, shared memory multiprocessors that isolate individual node failures are currently being developed. Even with these, because of the side effects of the cache coherency protocol, ...

8 Long-duration transaction support in design databases 100%
Waldemar Wieczerzycki

Proceedings of the fourth international conference on Information and knowledge management December 1995

9 Altruistic locking 100%

Kenneth Salem , Héctor García-Molina , Jeannie Shands
ACM Transactions on Database Systems (TODS) March 1994
Volume 19 Issue 1
Long-lived transactions (LLTs) hold on to database resources for relatively long periods of time, significantly delaying the completion of shorter and more common transactions. To alleviate this problem we propose an extension to two-phase locking, called altruistic locking, whereby LLTs can release their locks early. Transactions that access this released data are said to run in the wake of the LLT and must follow special locking rules. Like two-phase locking, altruistic locking is easy to ...

10 IBM's relational DBMS products: features and technologies 100%

C. Mohan
ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data June 1993
Volume 22 Issue 2
This paper very briefly summarizes the features and technologies implemented in the IBM relational DBMS products. The topics covered include record and index management, concurrency control and recovery methods, commit protocols, query optimization and execution techniques, high availability and support for parallelism and distributed data. Some indications of likely future product directions are also given.

11 Reducing recovery constraints on locking based protocols 100%

G. Alonso , D. Agrawal , A. El Abbadi
Proceedings of the thirteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems May 1994
Serializability is the standard correctness criterion for concurrency control. To ensure correctness in the presence of failures, recoverability is also imposed. Pragmatic considerations result in

further constraints, for instance, the existing log-based recovery implementations that use before-images warrant that transaction executions be strict. Strict executions are restrictive, thus sacrificing concurrency and throughput. In this paper we identify the relation between the recovery mecha ...

12 ARIES/IM: an efficient and high concurrency index management method using write-ahead logging 100%

C. Mohan , Frank Levine

ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

This paper provides a comprehensive treatment of index management in transaction systems. We present a method, called ARIESIIM (Algorithm for Recovery and Isolation Exploiting Semantics for Index Management), for concurrency control and recovery of B+-trees. ARIES/IM guarantees serializability and uses write-ahead logging for recovery. It supports very high concurrency and good performance by (1) treating as the lock of a key the same lock as the one on the ...

13 Efficient and flexible methods for transient versioning of records to avoid locking by read-only transactions 100%

C. Mohan , Hamid Pirahesh , Raymond Lorie

ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

We present efficient and flexible methods which permit read-only transactions that do not mind reading a possibly slightly old, but still consistent, version of the data base to execute without acquiring locks. This approach avoids the undesirable interferences between such queries and the typically shorter update transactions that cause unnecessary and costly delays. Indexed access by such queries is also supported, unlike by the earlier methods. Old versions of records are maintained only ...

14 Using delayed commitment in locking protocols for real-time databases — 100%

D. Agrawal , A. El Abbadi , R. Jeffers

ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992

Volume 21 Issue 2

In this paper, we propose locking protocols that are useful for real-time databases. Our approach is motivated from two main observations. First, locking protocols are widely accepted and used in most database systems. Second, in real-time databases it has been shown that the blocking behavior of transactions in locking protocols results in performance degradation. We use a new relationship between locks called ordered sharing to eliminate blocking that arises in the traditional locking pro ...

15 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging — 100%

C. Mohan , Don Haderle , Bruce Lindsay , Hamid Pirahesh , Peter Schwarz

ACM Transactions on Database Systems (TODS) March 1992

Volume 17 Issue 1

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

16 An approach to eliminate transaction blocking in locking protocols — 100%

D. Agrawal , A. El Abbadi , R. Jeffers

Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems July 1992

Results 1 - 16 of 16 short listing

PORTAL THE ACM DIGITAL LIBRARY

> home > about > feedback > log

US Patent & Trademark Office

Try the *new* Portal design
Give us your opinion after using it.

Search Results

Search Results for: **[((savepoint or save?point or "save point" or (partial <near/1> (rollback or roll?back))) and transaction and ((repeatable <near/1> read) or (lock$3 <sentence> (read or maintain* or duration) <sentence> (commit or committed or committing))))<AND>(meta_published_date <= 06-01-2001)]**
Found **9** of **127,132 searched.**

Search within Results

> Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score Binder

Results 1 - 9 of 9 short listing

1 Failure isolation and recovery in composite multidatabases 100%
Dexter P. Bradshaw
Proceedings of the 1994 conference of the Centre for Advanced Studies on Collaborative research October 1994
Most concurrency control schemes for guaranteeing global serializability in composite multidatabase systems are susceptible to rollbacks. Conservative schemes generate rollbacks because of transaction timeouts, while those of optimistic schemes are caused by certification failures. Typically, rollbacks on any branch of a flat distributed transaction cause a global abort. Global aborts during multidatabase composition degrade performance because of a waste of resources and reductions in multidata ...

2 Concurrency control: methods, performance, and analysis 100%
Alexander Thomasian
ACM Computing Surveys (CSUR) March 1998
Volume 30 Issue 1

3 ARIES/IM: an efficient and high concurrency index management method using write-ahead logging 100%
C. Mohan , Frank Levine
ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data June 1992
Volume 21 Issue 2
This paper provides a comprehensive treatment of index management in transaction systems. We present a method, called ARIESIIM (Algorithm for Recovery and Isolation Exploiting Semantics for Index Management), for concurrency control and recovery of B+-trees. ARIES/IM guarantees serializability and uses write-ahead logging for recovery. It supports very high concurrency and good performance by (1) treating as the lock of a key the same lock as the one on the ...

4 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging 100%
C. Mohan , Don Haderle , Bruce Lindsay , Hamid Pirahesh , Peter Schwarz
ACM Transactions on Database Systems (TODS) March 1992
Volume 17 Issue 1
DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

5 An approach to eliminate transaction blocking in locking protocols 100%
D. Agrawal , A. El Abbadi , R. Jeffers

Proceedings of the eleventh ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems July 1992

6 Efficient concurrency control in multidimensional access methods 100%
Kaushik Chakrabarti , Sharad Mehrotra
ACM SIGMOD Record , Proceedings of the 1999 ACM SIGMOD international conference on Management of data June 1999
Volume 28 Issue 2
The importance of multidimensional index structures to numerous emerging database applications is well established. However, before these index structures can be supported as access methods (AMs) in a “commercial-strength” database management system (DBMS), efficient techniques to provide transactional access to data via the index structure must be developed. Concurrent accesses to data via index structures introduce the problem of protecting ranges specified in the retrieval fr ...

7 Concurrency and recovery in generalized search trees 100%
Marcel Kornacker , C. Mohan , Joseph M. Hellerstein
ACM SIGMOD Record , Proceedings of the 1997 ACM SIGMOD international conference on Management of data June 1997
Volume 26 Issue 2
This paper presents general algorithms for concurrency control in tree-based access methods as well as a recovery protocol and a mechanism for ensuring repeatable read. The algorithms are developed in the context of the Generalized Search Tree (GiST) data structure, an index structure supporting an extensible set of queries and data types. Although developed in a GiST context, the algorithms are generally applicable to many tree-based access methods. The concurrency control protocol is base ...

8 IBM's relational DBMS products: features and technologies 99%
C. Mohan
ACM SIGMOD Record , Proceedings of the 1993 ACM SIGMOD international conference on Management of data June

1993
Volume 22 Issue 2
This paper very briefly summarizes the features and technologies implemented in the IBM relational DBMS products. The topics covered include record and index management, concurrency control and recovery methods, commit protocols, query optimization and execution techniques, high availability and support for parallelism and distributed data. Some indications of likely future product directions are also given.

9 DLFM: a transactional resource manager 80%
Hui-I Hsiao , Inderpal Narang
ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data May 2000
Volume 29 Issue 2

The DataLinks technology developed at IBM Almaden Research Center and now available in DB2 UDB 5.2 introduces a new data type called DATALINK for a database to reference and manage files stored external to the database. An external file is put under a database control by “linking” the file to the database. Control to a file can also be removed by “unlinking” it. The technology provides transactional semantics with respect to linking or unlinking the file when DATALINK ...

Results 1 - 9 of 9 short listing